

**ARCHITECTURE AND METHOD FOR PERFORMING A
FAST FOURIER TRANSFORM AND OFDM RECEIVER EMPLOYING THE SAME**

Inventor: Manish Goel
4417 Helston Drive
Plano, Texas 75024
County: Collin
Citizenship: India

Assignee: Texas Instruments Incorporated
P.O. Box 655474
MS 3999
Dallas, Texas 75265

CERTIFICATE OF EXPRESS MAIL	
I hereby certify that this correspondence, including the attachments listed, is being deposited with the United States Postal Service, Express Mail - Post Office to Addressee, Receipt No. <u>EV 3162162517 US</u> in an envelope addressed to Commissioner for Patents, Alexandria, VA22313, on the date shown below.	
<u>2/16/04</u> Date of Mailing	<u>DEBBIE SZNAJDER</u> Typed or printed name of person mailing
	<u><i>Debbie Szajder</i></u> Signature of person mailing

Hitt Gaines, P.C.
P.O. Box 832570
Richardson, Texas 75083
(972) 480-8800

**ARCHITECTURE AND METHOD FOR PERFORMING A
FAST FOURIER TRANSFORM AND OFDM RECEIVER EMPLOYING THE SAME**

CROSS-REFERENCE TO PROVISIONAL APPLICATION

[0001] This application claims the benefit of U.S. Provisional Application No. 60/450305 entitled "A High-Speed Scalable Architecture for FFT" to Manish Goel, filed on February 27, 2003, and incorporated herein by reference.

TECHNICAL FIELD OF THE INVENTION

[0002] The present invention is directed, in general, to fast Fourier transformation and, more specifically, to an FFT architecture, method of performing an FFT and an OFDM receiver employing the same.

BACKGROUND OF THE INVENTION

[0003] Communication systems extensively employ digital signal processing techniques to accomplish increasingly more sophisticated and complex computational algorithms. Expanding applications are being fueled by new technologies and increasing demand for products and services. The Discrete Fourier (or Frequency) Transform (DFT)

is employed in many of these applications to provide a needed transformation between sampled time-domain signals (that are usually digitized) and their frequency-domain equivalents. The DFT may be calculated in three different ways. A set of simultaneous equations can be employed, but this technique is too inefficient to be of practical use. Correlation techniques can also be used, but computational requirements make this technique cumbersome or expensive to implement on a broad scale.

[0004] The Fast Fourier (or Frequency) Transform (FFT) is an ingenious algorithm first discovered by Karl Friedrich Gauss, the great German mathematician of a century ago, and rediscovered and applied by J. W. Cooley and J. W. Tukey in 1965. The FFT is typically hundreds of times faster than the other DFT methods mentioned above and is therefore the algorithm of choice for a broad spectrum of applications employing the DFT. For example, the FFT is a critical element of a digital communication system that employs Orthogonal Frequency Division Multiplexing (OFDM) or Discrete Multitone (DMT) techniques.

[0005] The FFT is based on a "divide and conquer" model that decomposes a DFT into N points, which actually correspond to N separate DFTs consisting of a single point. The whole transform is then obtained from these simpler transforms. For example, an N -point DFT computation can be divided into two $N/2$ -point DFT computations that can be further divided into two $N/4$ -point DFT

computations, and so on until complete. Actually, the division occurs after a reorganization of the points, such that each point corresponds to a two-point DFT in each position when using a method based on radix-2. After this division and DFT computation, a merging process is performed in which the simpler DFT transforms are reassembled into the complete DFT transform. A basic computational element employed in the FFT is called a butterfly structure, which accepts two complex input numbers and performs one complex multiplication, one complex addition and one complex subtraction to produce two complex output numbers.

[0006] Pipelined FFT processors represent a specialized class of architectures for application specific, real-time DFT computation that use these fast algorithms and butterfly structures. They are characterized by continuous processing that employs a processor clock, synchronized with input data sampling, to produce one output sample for each processor clock cycle. Architectures for pipelined FFT processors have been the subject of intensive research as the demand for real-time processing has increased. This effort has resulted in several architectures that offer varying degrees of complexity, memory size, control requirements and utilization efficiencies. Unfortunately, these architectures usually require a close synchronization between input data sampling and processor clock rate, which can limit their breadth of application.

[0007] Accordingly, what is needed in the art is a new FFT

architecture that allows a wider disparity between input data sampling and processing clock rate.

SUMMARY OF THE INVENTION

[0008] To address the above-discussed deficiencies of the prior art, the present invention is directed to an FFT architecture. In one embodiment, the FFT architecture includes a pipeline segment having a plurality of data-independent pipelines that receive different time-domain data samples and generate therefrom corresponding intermediate results. Additionally, the FFT architecture also includes a parallel segment, coupled to all of the pipelines, that receives the corresponding intermediate results and generates therefrom corresponding frequency-domain results.

[0009] In another aspect, the present invention provides a method of performing an FFT. The method includes initially receiving different time-domain data samples into a plurality of data-independent pipelines of a pipeline segment, the data-independent pipelines generating therefrom corresponding intermediate results. The method also includes subsequently receiving the corresponding intermediate results into a parallel segment coupled to all of the pipelines, the parallel segment generating therefrom corresponding frequency-domain results.

[0010] Since Orthogonal Frequency Division Multiplexing (OFDM) is an advantageous application for FFT, the present invention also provides, in yet another aspect, an OFDM receiver. The OFDM receiver includes an input section that is coupled to a receive

antenna and an FFT section that is coupled to the receive section. The FFT section includes a pipeline segment having a plurality of data-independent pipelines that receive different time-domain data samples and generate therefrom corresponding intermediate results. The FFT section also includes a parallel segment, coupled to all of the pipelines, that receives the corresponding intermediate results and generates therefrom corresponding frequency-domain results. The OFDM receiver also includes an output section that is coupled to the FFT section.

[0011] The foregoing has outlined preferred and alternative features of the present invention so that those skilled in the art may better understand the detailed description of the invention that follows. Additional features of the invention will be described hereinafter that form the subject of the claims of the invention. Those skilled in the art should appreciate that they can readily use the disclosed conception and specific embodiment as a basis for designing or modifying other structures for carrying out the same purposes of the present invention. Those skilled in the art should also realize that such equivalent constructions do not depart from the spirit and scope of the invention in its broadest form.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] For a more complete understanding of the present invention, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, in which:

[0013] FIGURE 1 illustrates a system diagram of an embodiment of an OFDM transmitter/receiver pair constructed in accordance with the principles of the present invention;

[0014] FIGURE 2 illustrates a system diagram of an embodiment of a generalized, N-point pipeline/parallel FFT architecture constructed in accordance with the principles of the present invention;

[0015] FIGURE 3 illustrates a system diagram of an embodiment of a 256-point pipeline/parallel FFT architecture constructed in accordance with the principles of the present invention;

[0016] FIGURE 4 illustrates a system diagram of an embodiment of a 64-point FFT pipeline that may be employed in the pipeline segment of FIGURE 3;

[0017] FIGURE 5 illustrates an exemplary dataflow graph for a radix-2² FFT architecture wherein a 16-point transformation is shown for simplicity; and

[0018] FIGURE 6 illustrates a dataflow diagram for an embodiment of a 4-point, radix-2 FFT parallel segment that may be employed in the FFT parallel segment 310 of FIGURE 3.

DETAILED DESCRIPTION

[0019] As previously stated, FFT finds advantageous use in OFDM receivers. Accordingly, the overall architecture of an OFDM receiver will now be described. Referring initially to FIGURE 1, illustrated is a system diagram of an embodiment of an Orthogonal Frequency Division Multiplex (OFDM) transmitter/receiver pair, generally designated 100, constructed in accordance with the principles of the present invention. The OFDM transmitter/receiver pair 100 includes an OFDM transmitter 105 and an OFDM receiver 130. The OFDM transmitter 105 includes a transmitter input 106, a transmitter input section 110, a transmitter transform section 115, a transmitter output section 120 and a transmit antenna 124. The OFDM receiver 130 includes a receive antenna 131, a receiver input section 135, a pipeline/parallel fast Fourier transform (FFT) section 140, a receiver output section 145 and a receiver output 148.

[0020] The transmitter input section 110 includes a transmit forward error correction (FEC) stage 111, coupled to the transmitter input 106, and a quadrature amplitude modulation (QAM) mapper stage 112. The transmitter transform section 115 includes an N-point, inverse fast Fourier transform (IFFT) stage 116. The transmitter output section 120 includes a finite impulse response (FIR) filter stage 121, a digital-to-analog converter (DAC) stage

122 and a transmit radio frequency (RF) stage 123, which is coupled to the transmit antenna 124.

[0021] The receiver input section 135 includes a receive RF stage 136, which is coupled to the receive antenna 131, and an analog-to-digital converter (ADC) stage 137. The pipeline/parallel FFT section 140 includes a pipeline segment 141 and a parallel segment 142. The receiver output section 145 includes a QAM decoder stage 146 and a receive FEC stage 147, which is coupled to the receiver output 148.

[0022] The transmit FEC stage 111 provides forward error correction for a transmit input signal obtained from the transmitter input 106 and supplies an error-corrected input signal to the QAM mapper stage 112. The QAM mapper stage 112 codes the error-corrected transmit input signal for transmission and provides it to the IFFT stage 116. The N-point IFFT stage 116 transforms the error-corrected transmit input signal from the frequency domain to the time domain and supplies it to the FIR filter stage 121, where it is further filtered for transmission. The DAC stage 122 converts the transformed, filtered and error-corrected transmit input signal from a digital transmit signal to an analog transmit signal wherein it is further conditioned and modulated for transmission by the transmit RF stage 123 employing the transmit antenna 124.

[0023] The transmitted signal is received by the receive RF

stage 136 employing the receive antenna 131. This analog, time-domain receive signal is conditioned, demodulated and supplied to the ADC stage 137 wherein it is converted from an analog signal to a digital signal and supplied to the pipeline/parallel FFT section 140. The pipeline/parallel FFT section 140 transforms the received signal from the time domain to the frequency domain employing both the pipeline segment 141 and the parallel segment 142. The QAM decoder 146 decodes the transformed receive signal wherein it is forward error corrected by the FEC stage 147 and provided as a receive output signal from the receiver output 148.

[0024] The pipeline segment 141 performs an initial portion of the FFT and the parallel segment 142 uses this initial portion to complete the FFT. The pipeline segment 141 employs a plurality of data-independent pipelines that receive different time-domain data samples and uses them to generate corresponding initial portions of the FFT as intermediate results of the complete FFT. The parallel segment 142 is coupled to the outputs of the plurality of data-independent pipelines wherein it receives the corresponding intermediate results and employs them to generate the complete FFT. This pipelined and parallel arrangement advantageously allows an FFT to be performed efficiently even when the data sample rate exceeds the available system clock rate. Of course, other applications of the pipeline/parallel FFT section 140 are well within the broad scope of the present invention.

[0025] Having described one advantageous application for FFT, a novel FFT architecture will now be described. Accordingly, turning to FIGURE 2, illustrated is a system diagram of an embodiment of a generalized, N-point pipeline/parallel FFT architecture, generally designated 200, constructed in accordance with the principles of the present invention. The pipeline/parallel FFT architecture 200 provides an N-point FFT conversion. Generally, the pipeline/parallel FFT architecture 200 receives a parallelism level P of time-domain input samples and provides a parallelism level P of frequency-domain output samples for each clock cycle associated with the transformation.

[0026] The pipeline/parallel FFT architecture 200 includes a pipeline segment 205 and a parallel segment 210. The pipeline segment 205 includes a plurality of data-independent FFT pipelines 205a-205p that receive a plurality of different, parallel, time-domain input data samples x_a - x_p , respectively. Each of the FFT pipelines 205a-205p receives a single time-domain data sample at a time. The plurality of FFT pipelines 205a-205p generate a corresponding plurality of parallel intermediate results IR_a - IR_p , as shown. The parallel segment 210 accepts the parallel intermediate results IR_a - IR_p and generates a corresponding plurality of parallel, frequency-domain output samples X_a - X_p , each clock cycle. These pluralities correspond to the parallelism level P. The parallelism level P for a particular application is based

on both the time-domain data sample rate and the clock rate pertaining to the FFT application.

[0027] Turning now to FIGURE 3, illustrated is a system diagram of an embodiment of a 256-point pipeline/parallel FFT architecture, generally designated 300, constructed in accordance with the principles of the present invention. The pipeline/parallel FFT architecture 300 includes a pipeline segment 305 and a parallel segment 310. The pipeline segment 305 includes first, second, third and fourth 64-point FFT pipelines 305a, 305b, 305c, 305d, which are collectively designated as the FFT pipelines 305a-305d. In the illustrated embodiment, each of the FFT pipelines 305a-305d employs a radix-2² FFT pipeline structure, and the parallel segment 310 provides a 4-point (i.e., parallelism level of four), radix-2 parallel FFT structure.

[0028] Representative first, second, third and fourth time-domain data samples x_a , x_b , x_c , x_d are received by the FFT pipelines 305a-305d, respectively, and processed to provide respective first, second, third and fourth intermediate results IRa, IRb, IRc, IRd to the parallel segment 310. The second, third and fourth intermediate results IRb, IRc, IRd are weighted by first, second and third twiddle factors W1, W2, W3, and employed by the parallel segment 310 to provide first, second, third and fourth frequency-domain outputs Xa, Xb, Xc, Xd, as shown. Operation of each of these structures will be further described below.

[0029] Turning briefly to FIGURE 4, illustrated is a system diagram of an embodiment of a 64-point FFT pipeline, generally designated 400, that may be employed in the pipeline segment 305 of FIGURE 3. In the illustrated embodiment, the 64-point FFT pipeline 400 is implemented in hardware and includes first, second, third, fourth, fifth and sixth butterfly structures 405a, 405b, 405c, 405d, 405e, 405f, which are collectively designated as the butterfly structures 405a-405f, and first and second multipliers 410a, 410b. The 64-point FFT pipeline 400 (which is exemplary of any of the FFT pipelines 305a-305d) receives time-domain input data samples x_n at a pipeline input 401 and provides corresponding frequency-domain intermediate results IR_k at a pipeline output 402. The first and second multipliers 410a, 410b allow appropriate multiplication by first and second twiddle factors $W1(n)$, $W2(n)$. In the illustrated embodiment, the butterfly structures 405a-405f are radix- 2^2 single-path delay feedback architectures, which are well known in the pertinent art. Of course, other current or future developed pipeline architectures may be employed as appropriate to a particular application.

[0030] Turning briefly to FIGURE 5, illustrated is an exemplary dataflow graph for a radix- 2^2 FFT architecture, generally designated 500, wherein a 16-point transformation is shown for simplicity. The dataflow graph 500 employs 16 time-domain input data samples $x[0]$ - $x[15]$ and provides 16 frequency-domain transform

outputs $X[0]$ - $X[15]$, as shown. The dataflow graph 500 illustrates four exemplary butterfly dataflow areas BFI, BFII, BFIII, BFIV that correspond to four butterfly structures. A pipelined architecture is obtained by folding the dataflow graph 500 by a factor of $N/2$ (i.e., a factor of eight). It can be noted that such a folding will require a total of $\log_2 N$ (i.e., $\log_2 16$, or four) butterflies in pipelined architecture, as indicated. The third, fourth, fifth and sixth radix- 2^2 butterfly structures 405c, 405d, 405e, 405f of FIGURE 4 would provide such a structure. Additionally, the multiplicative operations in the dataflow graph 500 are such that only every other butterfly structure employs non-trivial multiplications involving the twiddle factors.

[0031] Turning briefly to FIGURE 6, illustrated is a dataflow diagram for an embodiment of a 4-point, radix-2 FFT parallel segment, generally designated 600, that may be employed in the FFT parallel segment 310 of FIGURE 3. The FFT parallel segment 600 receives frequency-domain intermediate results IRa - IRd from four pipelines and generates four frequency-domain outputs Xa - Xd during each clock time, employing the operations and dataflows shown.

[0032] Returning again to FIGURE 3, the 64-point FFT pipelines 305a-305d employ the radix- 2^2 architecture as discussed with respect to FIGURES 4 and 5. The radix- 2^2 architecture provides a multiplicative complexity equivalent to that of a radix-4 architecture while maintaining adder complexity and lower critical-

path properties of a radix-2 architecture, thereby facilitating pipeline implementation. The total number of complex multipliers in such a pipeline implementation is equal to $\log_4 N - 1$. There are also two complex adders per stage, thereby requiring a total of $2\log_2 N$ complex adders.

[0033] The number of complex multipliers and complex adders for the pipeline/parallel FFT architecture 300 is based on the N-points of the FFT architecture and the parallelism level P. The number of complex multipliers N_{cmult} is given by:

$$N_{\text{cmult}} = P(\text{the number of complex multipliers in an } N/P\text{-point pipelined FFT}) + (P-1) + (\text{the number of complex multipliers in a } P\text{-point parallel FFT}).$$

Now, assuming a radix-2² architecture for the pipelined N/P-point FFT and a radix-2 architecture for the parallel P-point FFT, the number of complex multipliers N_{cmult} may be expressed more concisely by:

$$N_{\text{cmult}} = P\log_4(N/P) - 1 \text{ where } P=1, 2, 4, \quad (1)$$

$$N_{\text{cmult}} = P\log_4(N/P) + 1 \text{ where } P=8, \text{ and} \quad (2)$$

$$N_{\text{cmult}} = P\log_4(N/P) + 9, \text{ where } P=16. \quad (3)$$

It is assumed that a 1, 2 or 4-point parallel FFT requires no complex multipliers while an 8 or 16-point FFT requires 2 or 10 complex multipliers, respectively.

[0034] Similarly, the number of complex adders N_{cadd} is given by:

$$N_{\text{cadd}} = P(\text{the number of complex adders in an } N/P\text{-point pipelined}$$

FFT)+(the number of complex adders in a P-point parallel FFT). Assuming a radix-2² architecture for the pipelined FFT and a radix-2 architecture for the parallel FFT, the number of complex adders N_{cadd} may be expressed more concisely by:

$$N_{cadd}=P(4\log_4N-2\log_4P), \quad (4)$$

where it is assumed that each N/P-point pipelined FFT requires $4\log_4(N/P)$ complex adders and that the P-point FFT requires $2\log_4P$ complex adders. Table 1 indicates the number of complex multipliers and adders required to implement an N-point FFT having a parallelism level P.

Table 1
Complex Multiplier-Adder Requirements
of Pipeline/Parallel Architectures

N	P	N_{cmult}	N_{cadd}
64	1	2	12
	2	5	22
	4	7	40
	8	17	72
128	1	3	14
	2	5	26
	4	11	48
	8	17	88
256	1	3	16
	2	7	30
	4	11	56
	8	25	104
512	1	4	18
	2	7	34
	4	15	64
	8	25	120
1024	1	4	20
	2	9	38
	4	15	72
	8	33	136

2048	1	5	22
	2	9	42
	4	19	80
	8	33	152

[0035] The gate complexity of the pipeline/parallel FFT architecture is strongly dependant upon the number of complex multipliers that are required. Strength reduction transformations are well known in the pertinent art and can be used to implement complex multiplications by employing three real multipliers and five real adders in place of four real multipliers and two real adders. Using a strength reduction transformation, an N-point pipeline/parallel FFT architecture having a parallelism level P of can be implemented with $3N_{\text{cmult}}$ real multipliers and $(2N_{\text{cadd}}+5N_{\text{cmult}})$ real adders. Table 2 indicates the number of real multipliers and adders required to implement an N-point FFT having a parallelism level P employing strength reduction.

Table 2
Real Multiplier-Adder Complexity
of Pipeline/Parallel Architectures
after Strength Reduction

N	P	Real Multipliers	Real Adders
64	1	6	34
	2	15	69
	4	21	115
	8	51	229
128	1	9	43
	2	15	77

	4	33	151
	8	51	261
256	1	9	47
	2	21	95
	4	33	167
	8	75	333
	16	151	667
512	1	12	56
	2	21	103
	4	45	203
	8	75	365
1024	1	12	60
	2	27	121
	4	45	219
	8	99	437
2048	1	15	69
	2	27	129
	4	57	255
	8	99	469

[0036] In summary, embodiments of the present invention directed to a pipeline/parallel FFT architecture, method of performing an FFT and an OFDM receiver employing the same have been presented. Advantages include allowing an FFT to be accomplished when a data sample rate exceeds an available system or transformation clock rate. The blending of pipeline and parallel FFT architectures provides an implementation trade-off between the complexity of an all parallel design and the constrained through-put of an all pipeline design as the number of points in an FFT conversion grows. Strength reduction transformations further allow a reduction in complex multiplications and additions.

[0037] Although the present invention has been described in detail, those skilled in the art should understand that they can

make various changes, substitutions and alterations herein without departing from the spirit and scope of the invention in its broadest form.